

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/822,535	03/30/2001	Erik S. Ruf	MS 160347.2/40062.120US01	1094
7590	08/18/2004		EXAMINER RUTTEN, JAMES D	
Homer L. Knearl Merchant & Gould P.C. P.O. Box 2903 Minneapolis, MN 55402-0903			ART UNIT 2122	PAPER NUMBER
DATE MAILED: 08/18/2004				

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b> 09/822,535	<b>Applicant(s)</b> RUF, ERIK S.	
	<b>Examiner</b> J. Derek Rutten	<b>Art Unit</b> 2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) ☒ Responsive to communication(s) filed on 25 June 2004.  
2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.  
3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) ☒ Claim(s) 1-23 is/are pending in the application.  
4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.  
5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.  
6) ☒ Claim(s) 1-23 is/are rejected.  
7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.  
8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) ☐ The specification is objected to by the Examiner.  
10) ☒ The drawing(s) filed on 30 March 2001 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).  
11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
a) ☐ All b) ☐ Some \* c) ☐ None of:  
1. ☐ Certified copies of the priority documents have been received.  
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)   | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftperson's Patent Drawing Review (PTO-948)                                    | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)             |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

### DETAILED ACTION

1. Acknowledgement is made of Applicant's amendment dated June 25, 2004, responding to the April 8, 2004, Office Action provided in the rejection of claims 1-23 wherein claims 10, 17, and 23 have been amended. Claims 1-23 remain pending in the application and have been fully considered by the examiner.
2. Applicant has primarily argued that the claims are patentable over Dattatri in view of Bacon because neither reference discloses a return constant table or optimized instructions as argued on page 10 of Applicant's response. This argument is not persuasive, as will be addressed under the *Response to Arguments* section below.
3. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

### *Response to Arguments*

4. Applicant's arguments filed June 25, 2004, have been fully considered but they are not persuasive. In the second paragraph on page 10, Applicant argues that Dattatri

Art Unit: 2122

does not generate a return constant table. While it is agreed that Dattatri does not generate a return constant table, Dattatri is not relied upon for this feature in the rejection of the claims. Therefore, this argument is moot. Applicant further argues in the second paragraph on page 10 that Dattatri does not generate an optimized instruction as required by the claim. However, Applicant points out that “the compiler has simply replaced the function call with a pointer to its corresponding vtbl entry.” This is indeed the optimization referred to by the examiner and generated by Dattatri. As the Applicant has admitted, “this substitution improves the efficiency of the function call”, and so can be interpreted as an optimized instruction.

In response to applicant's argument in the second paragraph of page 10 that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., obviation of function calls) are not recited in the rejected claims. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Applicant further argues in paragraph 3 on page 10, that Bacon does not generate a constant return table. However, as pointed out by the examiner on page 4 of the first Office action, Bacon does in fact generate a constant return table in the form of a cache table containing return values of side effect free procedures. Bacon essentially eliminates a method call by storing these return values for later retrieval. Applicant further argues that Bacon does not create an optimized instruction. However, Bacon is not relied upon for this limitation, so the argument is moot.

Art Unit: 2122

In response to applicant's argument in the third paragraph of page 10 that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., obviation of analysis of usage data) are not recited in the rejected claims. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

### ***Drawings***

5. Figure 8 should be designated by a legend such as --Prior Art-- because only that which is old is illustrated. See MPEP § 608.02(g). This requirement is maintained from the previous Office action of April 8, 2004. Applicant has argued that the figure depicts a computer system suitable for practicing the invention. As this figure depicts an exemplary system that is not new, and does not show incorporation of the invention, it should be labeled as --Prior Art--. Corrected drawings in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.121(d)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

***Claim Rejections***

6. Claims 1-23 stand finally rejected as unpatentable over prior art of record “C++: Effective Object-Oriented Software Construction” by Dattatri, in view of prior art of record “Compiler Transformations for High-Performance Computing” by Bacon et al. The claim rejections from the previous Office action dated April 8, 2004 is reproduced below for completeness.

***Claim Rejections - 35 USC § 103***

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

8. Claim 1-23 rejected under 35 U.S.C. 103(a) as being unpatentable over “C++: Effective Object-Oriented Software Construction” by Dattatri (hereinafter referred to as “Dattatri”), in view of “Compiler Transformations for High-Performance Computing” by Bacon et al. (hereinafter referred to as “Bacon”).

As per claim 1, Dattatri discloses:

*generating a table having an entry associated with the target method of the receiver object* (bottom of page 705: “Virtual functions are implemented through a table of pointers to functions, the vtbl.”); *and*

*generating an optimized instruction in association with the call site to retrieve a return value associated with the target method* (top of page 707: “This (pb->g()) can be written as ( \*(pb->vptr[2]) ) () ;”).

Dattatri does not expressly disclose the use of a return constant table encoded on computer program product.

However, in an analogous environment, Bacon teaches the construction of a cache to store the constant results of a side effect free procedure (pages 391-392, Section 6.8.9: “In such cases, it is possible to cache the results of recent invocations.”).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Dattatri’s method table with Bacon’s return constant cache table storage. One of ordinary skill would have been motivated to eliminate a method call in favor of directly accessing a constant value from a cache table. Method calls generate many extra instructions and computations, whereas a table lookup simply loads the value from memory. Further, it would have been obvious to one of ordinary skill in the art at the time the invention was made to encode the computer programs of Dattatri and Bacon on a computer program product. One of ordinary skill would have been motivated to encode a computer program as a means to distribute the program to a customer.

As per claim 2, the above rejection of claim 1 is incorporated. Dattatri does not expressly disclose inserting a constant return value.



However, Bacon teaches the insertion of a constant return value in a cache (pages 391-392, Section 6.8.9 as referenced above).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Bacon's method of constant value insertion into Dattatri's method table. One of ordinary skill would have been motivated to store a constant return value in a method table which provides fast access and to avoid having to execute a costly method call.

As per claim 3, the above rejection of claim 1 is incorporated. Dattatri discloses a dispatch table (bottom of page 705 as referenced above). Dattatri does not expressly disclose inserting a return value in a separate constant return table.

However, Bacon teaches the insertion of a constant return value in a separate cache (pages 391-392, Section 6.8.9 as referenced above).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Bacon's method of constant value insertion into Dattatri's method table. One of ordinary skill would have been motivated to store a constant return value in a separate table in order to maintain a simple data layout which also provides fast access while avoiding having to execute a costly method call.

As per claim 4, the above rejection of claim 3 is incorporated. Dattatri does not expressly disclose associating the return constant table with a receiver object.

However, Bacon teaches using a cached result with a procedure (page 391 Section 6.8.9 and page 392 Figure 51).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to associate Bacon's cache table with Dattatri's methods. One of ordinary skill would have been motivated to provide a means for accessing an improved data structure from a method that could benefit from using it.

As per claim 5, the above rejection of claim 1 is incorporated. Dattatri does not expressly disclose the determination of a non-transformable site and inserting into a dispatch table.

However, Bacon teaches the determination of a procedure as being non-transformable (page 391, Section 6.8.9, paragraph 1: "side-effect free"), and insertion of a return value into a cache (page 391 Section 6.8.9, paragraph 1).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Dattatri's method table with Bacon's non-transformable determination and insertion of return values in a cache. One of ordinary skill would have been motivated to ensure that the elimination of a method call was only made to a method that did not change the state of the program. Otherwise, the elimination of a method call would also eliminate whatever changes to state that would have occurred in the method.

As per claim 6, the above rejection of claim 1 is incorporated. Further, Dattatri discloses evaluating a plurality of possible target methods (bottom of page 705). Dattatri does not expressly disclose identification of return values.

However, Bacon teaches the identification of constant return values (pages 391-392, Section 6.8.9).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Dattatri's method table with Bacons identification of return values. One of ordinary skill would have been motivated to identify as many return values as possible in order to maximize the benefit of replacing a method call with access to stored values.

As per claim 7, the above rejection of claim 1 is incorporated. Dattatri does not expressly disclose the determination of side-effects.

However, Bacon teaches the application of function memoization only in the case when there are no side-effects (page 391 Section 6.8.9 paragraph 1).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Dattatri's method table with Bacon's side effect determination. One of ordinary skill would have been motivated to ensure that the elimination of a method call was only made to a method that did not change the state of the program. Otherwise, the elimination of a method call would also eliminate whatever changes to state that would have occurred in the method.

As per claim 8, the above rejection of claim 1 is incorporated. Dattatri does not expressly disclose an instruction for retrieval from the constant table.

However, Bacon teaches the generation of fetching instructions for retrieval of a return value from a data structure (page 392 Figure 51).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Dattatri's method table with Bacon's fetching instructions. One of ordinary skill would have been motivated to access return values stored in a table. Access to a table is generally faster than invoking a method call, and without instructions to access the table, it would be difficult to use.

As per claim 9, the above rejection of claim 1 is incorporated. Further, Dattatri discloses evaluating a plurality of possible target methods (bottom of page 705). Dattatri does not expressly disclose identification and storage of return values.

However, Bacon teaches the identification and storage of return values (pages 391-392, Section 6.8.9).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Dattatri's method table with Bacons identification and storage of return values. One of ordinary skill would have been motivated to identify as many return values as possible in order to maximize the benefit of replacing a method call with access to stored values.

As per claim 10, the above rejection of claim 1 is incorporated. Dattatri does not expressly disclose identification and optimization of control variables.

However, Bacon teaches constant propagation including:

*identifying a restricted set of one or more values of a control variable associated with a control operation (page 380 Figure 35: “n=64”);*

*identifying a restricted set of one or more types associated with the restricted set of one or more values of the control variable (Compilers inherently identify the types of variables used in the programs that they analyze. Without type identification, a compiler might attempt calculations with two incompatible pieces of data.); and*

*optimizing one or more control targets associated with the control operation based on the restricted set of one or more types (pages 379 and 380, Section 6.6.1 and Figure 35).*

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Bacon’s constant propagation in Dattatri’s object model. One of ordinary skill would have been motivated to precompute variables in a program in order to reduce computation at runtime, and also to reveal opportunities for other optimizations such as loop optimization.

As per claim 11, the above rejection of claim 1 is incorporated. Dattatri does not expressly disclose identification, mapping and optimization of control variables.

However, Bacon teaches constant propagation including:

*identifying a restricted set of one or more values associated with a control variable (page 380 Figure 35: “n=64”);*

*identifying one or more target methods providing the values associated with the restricted set (page 380 Figure 35);*

*mapping between the restricted set of values of the control variable and a restricted set of types based on the one or more target methods (Mapping values of variables and types or methods is an inherent operation of compilers. Without mapping, a compiler might attempt calculations with two incompatible pieces of data.); and*

*optimizing one or more control targets associated with the control statement based on the restricted set of types (pages 379 and 380, Section 6.6.1 and Figure 35).*

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Bacon’s constant propagation in Dattatri’s object model. One of ordinary skill would have been motivated to precompute variables in a program in order to reduce computation at runtime, and also to reveal opportunities for other optimizations such as loop optimization.

As per claim 12, Dattatri discloses:

*A method (page 701: “How Objects are Represented in C++”) comprising:*

*generating a table associated with the receiver object, the table having an entry associated with the target method of the receiver object (bottom of page*

Art Unit: 2122

705: "Virtual functions are implemented through a table of pointers to functions, the vtbl."); and

*generating an optimized instruction in association with the call site to retrieve a value associated with the target method* (top of page 707: "This (pb->g()) can be written as ( \*(pb->vptr[2]) ) ();").

Dattatri does not expressly disclose the determination of side effects or the use of a return constant table.

However, Bacon teaches the application of function memoization only in the case when there are no side-effects (page 391 Section 6.8.9 paragraph 1: "Memoization is an optimization that is applied to side-effect free procedures (that is, procedures that do not change the state of the program, also called *referentially transparent*)."). Bacon also teaches the construction of a cache to store the constant results of a side effect free procedure (pages 391-392, Section 6.8.9: "In such cases, it is possible to cache the results of recent invocations.").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Dattatri's method table with Bacon's side effect free constant storage. One of ordinary skill would have been motivated to ensure that the elimination of a method call was only made to a method that did not change the state of the program. Otherwise, the elimination of a method call would also eliminate whatever changes to state that would have occurred in the method. It is also beneficial to eliminate a method call in favor of directly

Art Unit: 2122

accessing a constant value from memory. Method calls generate many extra instructions and computations, whereas a table lookup simply loads the value from memory.

As per claims 13-21, the above rejection of claim 12 is incorporated. Further, all other limitations have been addressed in the above rejections of claims 2-4 and 6-11, respectively.

As per claims 22 and 23, all limitations have been addressed in the above rejections of claims 1 and 10, respectively.

### ***Conclusion***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to J. Derek Rutten whose telephone number is (703) 605-5233. The examiner can normally be reached on M-F 6:30-3:00.

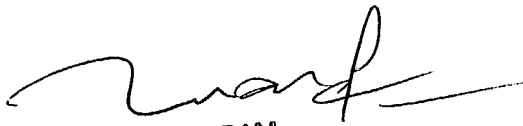
If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (703) 305-4552. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.



Art Unit: 2122

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

jdr



TUAN DAM  
SUPERVISORY PATENT EXAMINER